Journal of The Korea Society of Computer and Information
Vol. 19, No. 2, February 2014
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2014.19.2.067

# 클라우드 컴퓨팅 환경에서 이동노드 지원을 위한 기법

김 기 영*, 염 세 훈**

# Method for Mobile node in Cloud Computing Environments

Kiyoung, Kim*, Saehun, Yeom**

## 요 약

본 논문에서는 이동환경에서 이동단말이 핸드오프 시간과 오프로딩 시간을 측정하여 오프로딩의 수행을 판단하는 오프로딩 지연기법을 제안한다. 제안한 기법은 이동단말에서 핸드오프와 오프로딩 지연시간을 비교하여 오프로딩을 결정할 수 있도록 하여 고정노드를 대상으로 구현된 클라우드 컴퓨팅환경의 구조의 변경 없이 이동환경 클라우드 컴퓨팅을 지원한다. 효율성 분석을 위해 기존 연구에서 사용하는 서버와 단말의 에너지 소비측정을 사용하여 기존 방법과 에너지 소비를 비교 분석하였다. 모의실험 결과 오프로딩 지연 기법은 기존 방법보다 에너지 소비를 감소시키면서 유사한 작업수행 시간을 보이는 것을 확인하였다.

▸ Keywords : 클라우딩 컴퓨팅, 모바일 클라우딩, 모빌라우드, 오프로딩 지연, 핸드오프

## Abstract

In this paper, we proposed offloading delay method which determines effectively offloading timing by measuring of handoff delay and offloading time at mobile node side in mobile environment. The propose method measures each of handoff delay and offloading time and making decision of proper offloading timing on mobile node side. Therefore, it is possible to support cloud computing without changing previous implemented cloud computing structure for fixed node in a mobile environment. We compare the energy consumption of server and node to analyze efficiency of proposed method by using existing method of energy consumption measurement. Simulation results shows the reducing energy consumption more than previous method and operation time similar to previous method.

▸ Keywords : Cloud Computing, Mobile Cloud, Mobiloud, Offloading Delay, Handoff

---

•제1저자 : 김기영 •교신저자 : 염세훈
•투고일 : 2014. 2. 3, 심사일 : 2014. 2. 14, 게재확정일 : 2014. 2. 18.
* 서일대학교 컴퓨터소프웨어과(Dept. of Computer Software, Seoil University)
** 동서울대학교 컴퓨터소프트웨어과(Dept. of Computer Software, Dongseoul College)
본 연구는 2012년 서일대학교 교내학술연구비 지원에 의한 논문임

# I. 서 론

클라우드 컴퓨팅은 네트워크를 통해 원격지에 있는 컴퓨팅 자원에 접근하여 응용프로그램과 데이터를 활용하는 모델로 클라이언트에 응용프로그램 설치나 데이터의 저장 없이 업무나 프로그램 개발이 가능하다.

클라우드 컴퓨팅은 클라이언트에는 최소의 소프트웨어만을 설치하여 모바일 단말의 스토리지 비용과 모바일 단말에 데이터 저장을 하지 않음으로써 악의적인 데이터의 유출 방지가 가능하여 보안성이 높다. 또한 클라우드 컴퓨팅 서버에 응용프로그램을 설치하고 각각의 클라이언트에는 설치할 필요가 없어 기업에서 활용할 경우 프로그램 유지비용을 낮출 수 있다[1].

현재 제공되는 클라우드 컴퓨팅은 클라이언트의 데이터를 서버에 백업하는 서비스가 주를 이루고 있다. 이는 웹디스크, 네트워크 디스크와 유사한 것으로 PC에 저장된 데이터를 원격지 서버와 동기화하여 백업과 복구 작업을 쉽게 할 수 있도록 하는 서비스 형태로 제공되고 있다[2]. 클라우드 컴퓨팅은 이동환경을 고려하지 하지 않은 구조로 이동 단말을 지원하는 모바일 컴퓨팅 환경에 적용하면 효율적이지 못하다. 이동 환경은 단말이 지속적으로 이동을 하며 클라우드 서버에서 실행되고 있는 응용프로그램을 사용하는 특성을 갖기 때문이다. 기존 연구는 클라우드 서버 측의 에너지 소비를 고려하여 진행되었으며 최근에 이동 단말의 에너지 효율성에 대한 연구가 진행되고 있다[3,4].

기존 연구에서 에너지 사용은 두 단계로 구분하였다. 1단계는 이동 환경의 클라우드 컴퓨팅의 에너지 소비는 이동 단말의 작업을 수행에서 에너지 소비이며 2단계는 원격지의 클라우드 서버에 작업할 데이터의 송수신에 사용되는 에너지 소비로 정의 할 수 있다.

이동환경에서의 단말은 핸드오버의 수행에 따른 에너지 소비를 고려하여야 한다. 본 논문에서는 고정 환경만을 고려한 클라우드 컴퓨팅 구조에서 이동단말을 사용할 때 효율성을 높일 수 있는 방법을 제시하고 기존방법과 제안한 내용의 성능을 분석하고자 한다. 2장에서 관련연구를 3장에서 제안하는 내용을 4장에서는 제안 내용의 실험과 분석을 5장에서는 결론과 향후연구에 대해 기술한다.

# II. 관련 연구

## 1. 클라우드컴퓨팅

클라우드는 공공, 사설, 커뮤니티, 하이브리드 클라우드로 구분할 수 있다. 공공 클라우드는 인터넷에 접속된 모든 가입자가 사용할 수 있다. 컴퓨팅 자원은 웹을 통해 인터넷상으로 제공된다. 사용자들의 응용프로그램은 클라우드 서버와 스토리지 상에서 함께 실행된다. 사설 클라우드는 그룹을 구성하여 허가된 사용자만 그룹에 접속할 수 있는 형식이다. 물리적 구조는 해당 조직이나 서비스 제공자가 관리하고 소유한다.

커뮤니티는 2개 이상의 유사한 클라우드를 공유하는 방식이다. 하이브리 방식은  최소 2개의 클라우드를 조합하는 방식을 의미한다. 조합하는 클라우드는 공공, 사설, 커뮤니티가 될 수 있다[5,6].

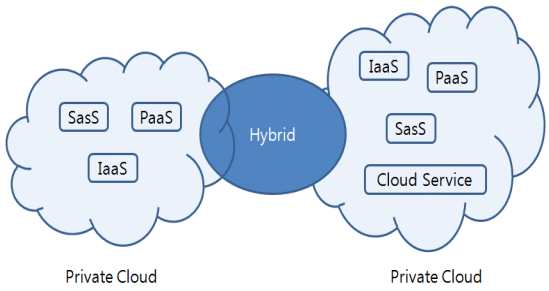


그림 1. 모빌 클라우드 구조
fig 1. Mobile Cloud Architecture

사용자가 가입할 수 있는 클라우드의 형식은 3가지가 존재하며 SaaS(Software as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service)이다[7,8]. 보안지원을 위해 가상회선 기반의 NaaS(Network as a Service)를 사용할 수 있다.

SaaS

사용자에게 소프트웨어 라이선스를 부여하여 소프트웨어 개발을 하는 모델로 클라이언트 사용자가 클라우드 서버에서 실행되는 어플리케이션을 사용할 수 있다. 클라우드 서버의 응용프로그램은 클라이언트 단말기의 웹브라우저와 같은 thin-client인터페이스를 통해 사용할 수 있다. 사용자는 네트워크, 서비스, 운영체제 스토리지 등을 포함하는 클라우드

구조를 제어할 필요가 없다.

PaaS

컴퓨팅 플랫폼과 솔루션 스택을 제공한다. 서비스 설정과 같은 소프트웨어 단계로 사용자가 개발에 필요한 요소들을 사용할 수 있도록 하는 형태이다. IaaS와 NasS는 하위수준으로 제한되어 PaaS는 API를 제공할 수 있도록 구성한다. 사용자에게 제공되는 기능은 서비스 제공자가 지원하는 프로그램언어와 툴을 사용하여 사용자가 생성한 응용프로그램이나 클라우드에 배치한다. 클라이언트 상의 사용자는 네트워크, 서버, 운영체제, 스토리지에 대한 제어나 관리를 하지 않는다는 점은 SasS와 같다.

PaaS는 WAMP(Windows, Apache, MySql, PHP), LAMP(Linux, Apache, MySql, PHP), XAMP(X-cross paltform)와 같은 운영체제와 응용서버의 조합을 제공한다.

IaaS

사용자가 외부의 스토리지와 자원을 사용할 수 있도록 하며 버튼업 형태로 제공하며 가상환경 플랫폼을 서비스로 제공한다. 한 개 이상의 VM을 계산, 통신, 저장을 수행하는 이동단말에 제공할 수 있다. 일반적으로 운영체제는 영상처리, 데이터마이닝 등을 수행하는 이동단말 상에서 구동하기에 적합하지 않다.

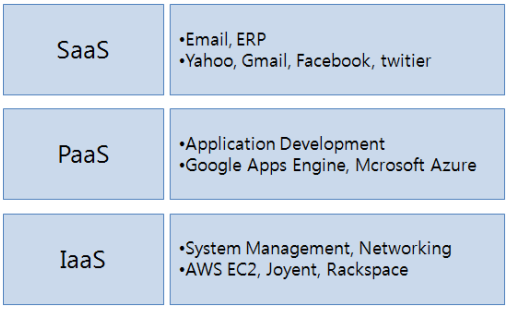


그림 2. 클라우드 서비스
fig 2. Cloud Services

NaaS

가상네트워크 환경에서 유연성과 보안 지원을 완벽하게 지원하지 못한다. 사적인 데이터는 신뢰할 수 있는 사용자와 공유할 수 있어야 한다. IaaS기반에서 송신자는 VM에게 데이터를 전송할 수 있다. 클라우드 체제에서 이동 단말은 VM을 제어하고 동작할 수 있기 때문에 전송되는 데이터를 보호할 필요가 있다. 클라우드 체제에서 VPN을 사용하면 보안이 가능하다. 클라우드는 VLAN을 adhoc형식으로 설정할 수 있도록 한다.

모빌클라우드 통신 세션을 지원하기 위해서는 충분한 대역폭을 보장하는 다수의 VLAN들을 제어할 수 있어야 한다. 이를 Network-as-a-Service(NaaS)라고 한다. VM간통신은 주로 클라우드 어플리케이션에서 사용되는 것으로 VLAN의 용량은 모비클라우드에서 중요한 요소로 원격컴퓨팅과 클라우드 어플리케이션 기반의 스토리지 서비스와 근본적으로 차이가 있다.

클라우드 컴퓨팅을 이동환경에서 활용 예는 [그림 3]과 같다. 클라우드 컴퓨팅은 이동 컴퓨팅을 고려하지 않은 구조로 설계되어 이동환경에서 클라우드 서버에 있는 응용프로그램을 사용하면 효율적이지 못하다.

클라우드 컴퓨팅의 효율성에 관한 연구는 일반적으로 데이터센터의 에너지 비용을 감소시키는 것을 주로 다루었으며 최근 들어 이동 단말의 에너지 소비에 관한 연구가 진행되고 있다[10-14].

또한 오프로딩 방법이 모바일 단말의 에너지를 절약할 수 있는지를 결정하는 연구도 진행 중이다. 이동단말에서 클라우드로 오프로딩한 수행할 작업이 종단간 에너지 소비를 감소할 수 있는지를 실험한 예는 없다. 한편 저성능의 클라이언트에 대한 에너지 소비에 관한 연구 분야도 있다.[15,16].

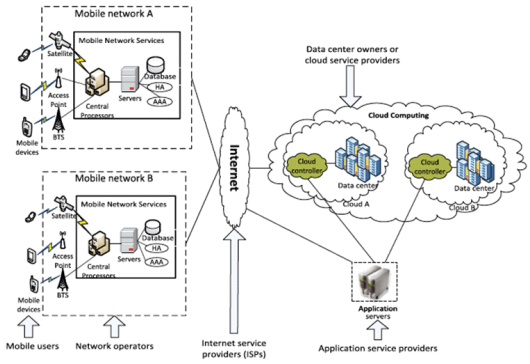


그림 3 . 모빌 클라우드 구조
fig 3. Mobile Cloud Architecture

이동환경에는 단말이 이동하게 되면 도착한 장소의 AP와 다시 연결을 재설정하여 통신을 끊김없이 지속시키는 핸드오프를 수행한다. 모바일 클라우드 컴퓨팅환경을 지원하기 위해서는 이동단말의 이동에 따른 핸드오프를 고려한 기법이 필요하다.

## 2. 모바일 컴퓨팅

〔그림 4〕는 무선환경에서 이동단말이 도착한 영역의 AP(Access Point)에 재연결하는 것을 보인다.

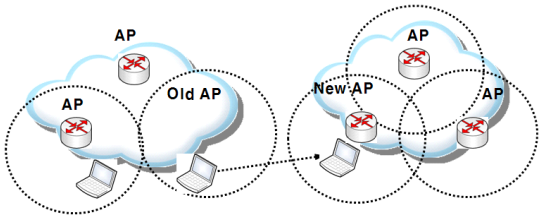


그림4. 무선환경에서의 핸드오프
fig 4. Handoff in Mobile Environment

이동단말은 이전에 위치한 공간에서 새로운 공간에 도착하면 핸드오프를 수행하여 원격지의 서버와 통신연결을 유지한다. 단말은 등록된 AP를 통해 원격 단말과 세션을 유지하여 통신을 한다. 단말이 B지역으로 이동하면 새로운 AP와 세션을 연결하여 기존의 원격 단말과 통신을 유지한다〔17〕.

핸드오프가 완료되기 전까지 통신 단절이 발생하지만 실시간 콘텐츠 등을 통신전체의 QoS의 영향을 미치지 않는다.

핸드오프 지연시간은 〔그림 5〕와 같이 이동한 단말이 새로운 AP의 채널을 탐색, 인증, 연결 시간으로 구성된다. 이동단말은 AP의 광고메시지를 수신하여 새로운 AP를 인식하고 사용 가능한 채널을 탐색하고 사용가능한 채널을 선택한다. 선택 후에 단말이 AP에 접속할 권한을 갖고 있는지 선택한 채널을 통해 인증절차를 수행한다. 인증을 요구를 설정하지 않은 AP는 이 단계를 생략한다. 이후 이동단말과 AP사이에 통신 재연결을 수행하여 끊김 없는 통신이 가능하게 한다. 이동단말의 통신지원을 위해 핸드오프 지연은 필연적으로 발생하기 때문에 지연에 민감한 실시간 콘텐츠를 수신할 때는 버퍼링이나 스트리밍 방법을 활용하여 통신단절로 발생하는 지연을 해결하고 있다.

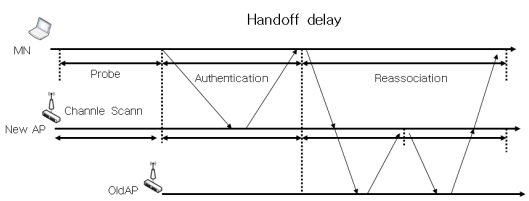


그림 5. 핸드오프 지연시간 구성요소
fig 5. Handoff Delay

## 3. 모빌라우드

모빌라우드는 thin-clinet 방식으로 thin-client는 소프트가 클라우드 서버에 작업요청을 하고 결과를 수신하여 처리하는 방식이다〔18〕.

모빌라우드는 클라우드 서비스와 모바일 서비스를 결합한 것으로 Sass, IaaS, Paas와 호환 기능을 갖고 있다. 모빌라우드는 이동성을 갖는 단말을 지원하는 개념으로 기존의 이동단말인 넷북, 노트북을 포함하여 thin-client를 지원한다. 클라우드 컴퓨팅이 단순히 이동환경으로 확장되는 것이 아닌 모바일 서비스를 클라우드 컴퓨팅의 구조에 포함시켜 이동 단말 사용자가 필요할 때 장소와 시간의 제약 없이 클라우드 컴퓨팅을 사용할 수 있도록 한다. 서비스 형태로 보면 공공, 사설 클라우드의 성격을 갖는다.

# III. 모바일 클라우딩 기법

## 1. 고정환경 클라우딩[19]

모바일 클라우드에서 단말의 이동에 따른 시나리오는 다음과 같다.

이동단말이 모든 작업을 완료 때까지 정지상태, 오프로딩 후에 이동, 오프로딩을 완료하고 이동하는 경우, 작업을 오프로딩 수행하기 전에 이동하는 경우, 로 구분할 수 있다.

첫 번째 오프로딩을 포함한 작업을 모두 수행할 때까지 이동하지 않는 경우는 기존 클라우드 컴퓨팅 환경과 동일한 조건이며 두 번째 오프로딩 후에 이동하는 경우와 세 번째 오프로딩을 완료하고 이동하는 경우 이동환경의 조건과 동일하다. 네 번째 오프로딩을 완료하기 전에 단말이 이동하는 경우는 클라우드 컴퓨팅에서 고려하지 않은 경우로 이동환경에서 클라우드 컴퓨팅에서 고려해야하는 부분이다. 오프로딩이 완료되지 않은 상태에서 단말이 이동하게 되면 클라우딩 서버에 작업내용이 전달되지 않은 상태이기 때문에 클라우딩 서버 쪽에서는 작업을 수행할 수 없고 단말이 이동하는 동안 작업수행은 지연되게 된다.

따라서 오프로딩이 완료되지 않은 상태에서는 클라이언트의 자원을 활용하여 작업을 수행하는 것을 고려할 수 있다.

클라우드 컴퓨팅의 효율성 분석은 에너지 소비를 측정하는 방식을 주로 사용한다. 클라우드 컴퓨팅의 에너지 소비는 클라이언트에서 수행할 작업의 실행을 위해 소비되는 에너지와 작업할 내용을 클라우드 서버에 전송하고 클라우드 서버에서

계산된 결과를 수신하는데 소비되는 송수신 에너지로 정의할 수 있다. 에너지를 측정하는 단위는 초 단위를 사용하며 작업을 수행하는데 필요한 에너지의 합으로 정의한다. 클라우드 서버를 사용하여 작업을 수행하면 이동 단말이 작업을 단독으로 수행할 때보다 에너지 소비를 감소할 수 있다.

고정 환경 클라우드 컴퓨팅에서 에너지 소비는 다음 수식과 같다.

작업을 수행하는 시간을 T, 사용된 전원을 P라고 하면 클라이언트에서 작업내용을 실행하기 위해 소비하는 에너지는 다음 수식과 같다.

$$Energy_{client} = P_{client} \times Time_{task}$$

클라우드 컴퓨팅은 클라이언트에서 요청한 응용프로그램을 실행하거나 데이터를 처리하여 클라이언트에 결과를 전송하는 방식이므로 서버와 클라이언트의 소비전력을 고려하여야 한다. 따라서 클라우드 컴퓨팅에서 소비되는 에너지는 다음 수식과 같다.

$$Energy_{server} = P_{client} \times P_{task'} + P_{server} \times P_{task''}$$

Ttask″≤Task′, and Ttask″는 서버가 작업을 수행하는데 소요되는 시간이며 Task′는 클라이언트의 작업수행 시간을 나타낸다. 작업의 내용을 클라우드 서버에 송수신하는 시간을 포함한다. 따라서 클라우드 컴퓨팅이 클라우드 서버에서 작업을 처리하는 것이 더 효율적이기 위해서는 클라이언트에서 처리하는데 소요되는 에너지 보다 작아야 한다.

따라서 $Energy_{server} \leq Energy_{client}$일을 만족할 때 클라우드 컴퓨팅을 사용하는 것이 더 효율적이다.

## 2. 이동환경 클라우딩

클라우드 컴퓨팅은 처리할 작업을 네트워크를 통해 클라우드 서버에 전송하고 결과를 수신하는 단계를 포함한다. 따라서 전체 에너지 소비는 다음 수식과 같이 표현할 수 있다.

$$Energy_{total} = Energy_{server} + Energy_{client} + Energy_{\text{network}}$$

$$Eenergy_{\text{network}} = (P_{\text{network}} \times Time_{\text{network}}) \times h$$

h는 클라이언트와 클라우드 서버 간에 홉의 개수를 의미한다. 이동환경 클라우드 컴퓨팅에서는 단말의 이동에 따른 핸드오프를 고려하여야한다. 핸드오프로 인해 고정 환경에서 발생하지 않는 지연으로 인해 추가적인 에너지 소비가 발생한다. 이동환경의 노드는 상황에 따라 세션이 종료될 때까지 정지상태를 유지할 수 있으며 이동상태일 때로 구분할 수 있다. 정지상태일 때는 고정 환경 클라우드와 동일한 조건이지만 단말이 세션동안 이동을 하는 상태에서는 단말이 세션유지를 위해 수행하는 핸드오프를 고려하여야 한다. 이동단말은 클라우드 컴퓨팅을 사용하지 않아도 통신 유지를 위한 핸드오프를 수행한다. 클라우드 컴퓨팅을 이동 단말이 사용할 때는 핸드오프에 따른 오버헤드를 고려 하야한다. 핸드오프를 고려한 이동단말의 평균 에너지 소비는 다음 수식과 같다.

$$Energy_{handoff} = P_{client}{}^{*}(P^{*}\, Time_{Task1}) + P_{client}t^{*}((1-P)^{*}time_{Task2})$$

$$Time_{task1} = k$$
$$Time_{task2} = k \times Frequency of Moving$$

P는 이동 단말이 정지해 있을 확률이며 TimeTask1은 정지해 있을 때 소요되는 시간, TimeTask2는 이동할 때 소요되는 시간 k는 핸드오프 수행시간 상수이다.

$$Energy_{total} = Energy_{server} + Energy_{client} + Energy_{\neq \text{twork}} + Energy_{handoff}$$

단말이 이동상태에서 현재 등록된 AP를 통해 오프로딩을 요청하고 이동하여 새로운 AP에 등록을 하면 추가적인 오버헤드는 발생하지 않는다. 하지만 오프로딩 요청 중에 핸드오프를 수행하게 되면 새롭게 도착한 AP에서 다시 오프로딩을 요청하여야 한다. 단말의 핸드오프 시간이 오프로딩 수행에 필요한 시간보다 크다면 이동단말은 클라우드 서버에 자신의 작업을 요청하여 수행할 수 없게 된다. 핸드오프 시간은 평균값을 갖지만 이동속도가 일정 속도 이상이면 핸드오프 수행 시간동안 해당 AP와 연결 설정을 할 수 없다. 따라서 이동단말의 상태에 따라 정지상태와 이동상태로 〈표 1〉과 같이 정의하며 이동상태는 오버로딩 가능상태와 오버로딩 불가능 상태로 다시 정의한다. 이동상태에서 오프로딩 가능 상태는 이동하는 단말이 자신이 처리해야할 작업을 클라우드 서버에 요청을 완료하였지만 결과를 수신하지 못하는 상태를 의미한다. 이 경우 이동단말의 작업 요청을 클라우드 서버에서 작업을 처리한 상태이므로 이동 단말이 클라우드 서버에서 작업한 결과를 수신하지 못하는 지연이 발생한다. 고정환경과 달리 단말이 이동을 하는 상태에서는 핸드오프를 통해 통신연결을 유지하지 않으

면 클라우드 서버에서 작업결과를 수신할 수 없다.

표 4 이동단말의 오프로딩 상태
table 4. Offloading State

| 상태 | 오프로딩 | 비 고 |
|---|---|---|
| 정지상태 | 오프로딩 가능 | offloading time < handoff time |
| 이동상태 | 오프로딩 가능 | offloading time < handoff time |
| | 오프로딩 불능 | offloading time > handoff time |

불능상태의 경우 핸드오프 시간보다 오프로딩 시간이 작아지기 전까지 이동단말은 자신의 작업을 클라우드 서버에 전송할 수 없다. 현재 클라우드 컴퓨팅 구조를 사용하면 이동단말은 오프로딩을 계속해서 수행하게 되고 에너지 소모가 발생한다. 이를 해결하기 위해서 이동단말은 클라우드 서버에 작업요청 전송을 하지 않음으로써 에너지 사용을 감소시킬 수 있다. 이동단말은 오프로딩이 가능한지를 판단하고 가능한 경우 오프로딩을 수행하여 에너지 효율을 높일 수 있다. 오프로딩 유무를 판단하기 위해 이동단말은 자신이 가입한 클라우드 서버로 작업요청에 소요되는 평균 시간을 사용한다. 오프로딩 평균 시간은 이전 작업을 통해 수집할 수 있다. 오프로딩에 필요한 시간은 클라우드 서버에 작업할 내용을 전송하는데 소요되는 시간으로 네트워크상의 전송시간에 해당한다. 클라우드 컴퓨팅은 클라이언트의 작업내용을 클라우드 서버에서 처리하는 방식이기 때문에 작업을 처리하는 시간은 고려하지 않고 전송속도만을 고려하면 된다.

이동단말은 이전에 수행한 작업들의 평균 작업요청시간과 평균 핸드오프 지연시간을 계산하여 새롭게 도착한 AP에서의 오프로딩의 가능성을 판단한다. 오프로딩을 할 수 없는 핸드오프 지연상태가 되면 이동단말은 오프로딩 요청을 하지 않는다. 이후 핸드오프 지연시간이 오프로딩 시간보다 작거나 같아지면 오프로딩을 수행하여 클라우드 서버에서 수행한 작업결과를 수신한다.

## III. 실험 및 결과

제안한 기법을 효율성을 분석하기 위해 모의실험을 수행하였다. 기존 고정환경 클라우드 기법과 제안한 이동환경을 고려한 클라우드 기법에서 작업처리시간, 에너지 소모를 기준으로 분석하였다. 이동상태에서 오프로딩이 가능한 경우는 기존 클라우드 기법과 차이가 없어 오프로딩이 불가능 상태만을 비교하였다.

### 1. 실험환경

제안한 클라우드 컴퓨팅 환경에서 이동노드 지원을 위한 기법의 성능평가를 위해 가정한 모의실험 환경은 [그림 6]과 같다. 다수의 AP가 존재하는 경우 이동 단말이 도착한 네트워크의 모든 AP의 신호와 채널을 탐색하는 시간을 제외하기 위해 한 개의 네트워크에는 한 개의 AP가 존재하는 단순화한 네트워크를 가정하였으며 이동 단말이 이동하는 네트워크의 수를 Ns로 정의하고 각각의 네트워크에는 한 개의 AP가 존재하는 것으로 하였다. 핸드오프가 발생하는 확률을 0.8-1까지 발생시켜 각각의 확률에서 소요되는 에너지 값을 비교하였다. 이동단말과 클라우드 서버에서 소비되는 에너지는 제안한 기법과 기존 방법에 동일하게 적용하였다.

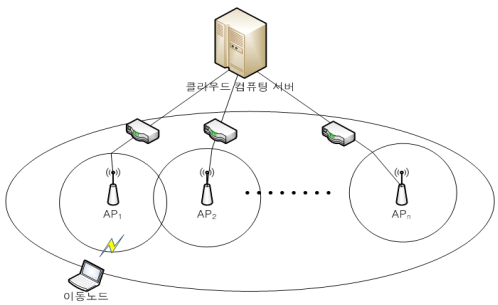


그림 6. 실험환경
fig 6. Experimental Environment

모의실험에서 사용한 변수와 의미는 <표 2>와 같다.

표 5. 실험변수
table 5. Experimental Values

| 변수 | 의 미 | 값 |
|---|---|---|
| NS | 외부네트워크의 수 | 10 |
| Hp | Handoff시 < offloading 확률 | 0.8-1 |
| Ep | 핸드오프 후 에러 발생 확률 | 0 |
| Offpw | offloading energy | 1.5ms |
| HOpw | handoff energy | 10 |
| Ndelay | 네트워크 지연시간 | 5ms |

## 2. 실험 결과 분석

제안한 기법의 성능평가를 위해 모의실험을 수행하였으며, 모의실험 환경은 다음과 같다.

네트워크의 집합을 Ns라하고 Ns = { SN1, SN2, … SNn}으로 정의한다. 이전의 오프로딩 시간과 핸드오프 시간을 비교하여 오프로딩을 결정은 핸드오프 시간이 오프로딩 시간이 작을 확률로 처리하였다.

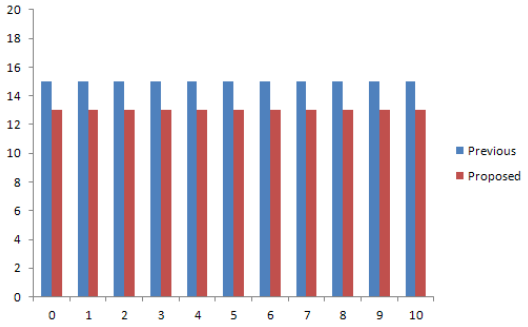


그림 7. 작업처리 시간
fig 7. Processing Time

〔그림 7〕은 단말의 이동에 따른 작업처리 시간을 나타낸다. X축은 단말의 이동 횟수, Y축은 처리시간을 나타낸다. 오프로딩 시간을 포함한 작업처리 시간으로 오프로딩 수행한 시간만큼 작업처리가 지연되는 것을 알 수 있다. 기존 방법은 단말의 이동을 고려하지 않고 클라우드 컴퓨팅 처리를 하도록 하였다. 따라서 핸드오프가 발생하여 오프로딩이 중단되는 경우 다시 처리하게 된다. 따라서 제안한 기법보다 수행시간이 길어지는 것을 알 수 있다. 제안한 기법은 오프로딩을 할 수 없는 경우에 핸드오프 시간이 오프로딩 시간보다 작아질 때 까지 오프로딩 작업을 지연시켜 불필요한 시도를 제거하였다. 오프로딩 수행 횟수를 줄이지만 클라우드 컴퓨팅은 오프로드가 수행된 이후 클라우드 서버에서 작업을 처리하기 때문에 전체 작업시간에 있어서는 기존 방법과 유사하다.

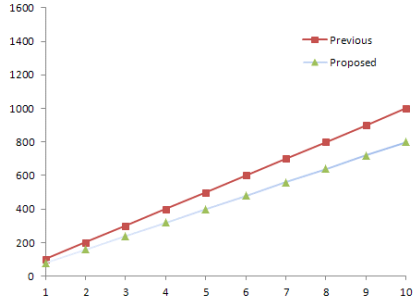


그림 8. 이동 횟수에 따른 에너지 소비
fig 8. Energy Consumption per Moving

〔그림 8〕은 단말의 이동 횟수에 따른 에너지 소모를 나타낸다. X축은 이동 횟수를 Y축은 에너지 소비를 나타낸다. 이동 횟수가 증가할수록 에너지 소비도 증가하는 것을 알 수 있다. 이는 이동에 따른 핸드오프와 오프로딩의 재시도로 인한 것으로 오프로딩 수행 중 이동을 하여 다시 오프로딩을 수행하는 기존 방법에서 에너지 소모가 더 크다. 이동 단말이 새롭게 도착한 네트워크에서 오프로딩 시간보다 핸드오프 시간이 작아지면 기존 방법도 동일한 에너지 소모를 보이지만 이동 속도를 일정시간 유지하는 이동 단말의 특성상 1회 이내에 이동속도가 낮아지는 확률은 낮다.

〔그림 9〕는 오프로딩 작업수행 시간에 따른 에너지 소모를 나타낸다.

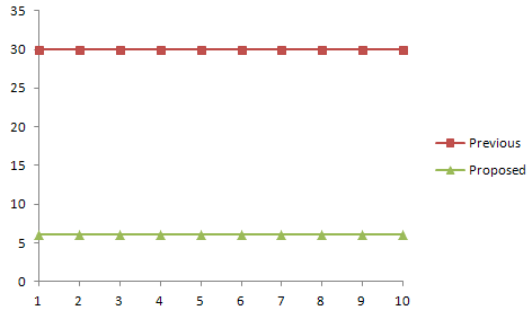


그림 9. 오프로딩 에너지 소비
fig 9. Energy for Offloading Consumption

제안한 기법은 핸드오프 시간이 오프로딩 시간보다 작은 경우에는 오프로딩을 수행하지 않기 때문에 낮은 에너지 소모를 보였다. 이에 반해 기존 클라우드 기법의 경우에는 이동과 관계없이 오프로딩을 수행하기 때문에 오프로딩에 필요한 에너지를 지속적으로 소비하는 것을 알 수 있다.

따라서 낮은 컴퓨팅 성능과 배터리를 갖는 이동단말에서

지속적인 오프로딩 시도는 클라이언트의 사용시간을 단축시키게 된다. 반면 제안한 기법은 오프로딩이 가능할 때까지 오프로딩 시도를 지연시켜 클라이언트의 배터리 사용시간을 개선할 수 있다. [그림 10]은 오프로딩을 할 수 없는 상태에서 지속적으로 핸드오프와 오프로딩에 따른 소비 에너지를 누적한 결과를 보인다. X축은 단말의 이동횟수, Y축은 소비된 에너지의 누적을 나타낸다.

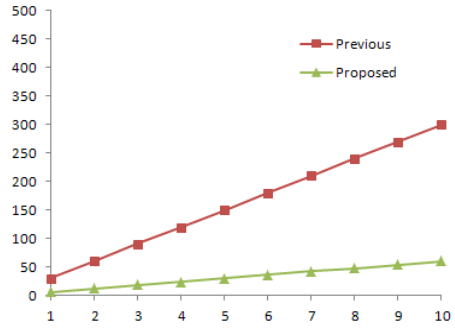


그림 10. 오프로딩 에너지 소비(누적)
fig 10. Energy for Offloading Consumption(Accumulation)

노드가 고속으로 이동하는 상태를 유지할 때 기존 방법의 클라우드 컴퓨팅은 노드의 배터리 소모를 발생시킨다. 반면 제안한 기법은 핸드오프 수행으로 인한 배터리 소모만을 하여 고속으로 이동하는 노드에서 더 효율적이라는 것을 알 수 있다.

## IV. 결론

클라우드 컴퓨팅은 클라이언트에 응용프로그램을 설치하지 않고 클라우드 서버에 설치되어 있는 응용프로그램을 이용하여 작업을 처리하거나 클라우드 서버의 저장 공간을 사용하는 방식으로 네트워크의 고속화로 다양한 서비스가 지원되고 있다. 하지만 현재 이동환경에서 지원하고 있는 서비스는 스토리지 서비스로 한정되어 있다. 클라우드 컴퓨팅에서 클라우드 서버에 설치되어 있는 응용프로그램을 사용하기 위해서 클라이언트의 작업내용을 클라우드 서버에 전송하고 실행하는 오프로딩을 수행하여야 한다.

본 논문에서는 고정환경과 달리 이동환경에서는 통신을 지속하기 위한 핸드오프에 따른 제약을 고려하여 이동환경에서 클라우드 컴퓨팅을 효율적으로 사용할 수 있는 기법을 제시하였다. 제안한 기법은 오프로딩 수행을 이동단말의 평균 핸드오프 시간과 평균오프로딩 시간을 기반으로 오프로딩 수행을 판단하여 상태에 따라 오프로딩 지연을 통해 클라이언트의 에너지 효율을 감소시켰다. 오프로딩 지연으로 인해 클라이언트의 전체 수행시간은 기존 방법과 비교하였을 때 클라이언트의 에너지 효율은 높아졌으며 동일한 수행시간을 보였다.

향후 연구로는 오프로딩 지연 기법을 효율적으로 지원할 수 있는 결정 알고리즘을 연구와 오프로딩을 부분적으로 처리하여 클라이언트의 에너지 효율을 감소시키면서 전체 작업수행 시간을 감소시킬 수 있는 연구가 필요하다.

## 참고문헌

[1] Amrhein, D. & Willenborg, R., "Cloud computing for the enterprise," Part 3: Using, Apr. 2009.
[2] Ramesh Jain. Quality of experience, IEEE Multimedia, vol. 11, Jan. 2004.
[3] Le Guan, Xu Ke, Meina Song, and Junde Song, "A Survey of Research on Mobile Cloud Computing," IEEE/ACIS 10th International Conference on Computer and Information Science (ICIS), pp. 387-392, May 2011.
[4] Xiaopeng Fan, Jiannong Cao, and Haixia Mao. "A Survey of Mobile Cloud Computing," ZTE Communications, no. 9-1 pp. 4-8, Mar 2011.
[5] R.J. Bayardo, and R. Srikant, "Technological Solutions for Protecting Privacy," IEEE Computer, no. 36-9, pp. 115-118, Sept. 2003.
[6] Chetan S., Gautam Kumar, K. Dinesh, Mathew K. and Abhimanyu M.A., "Cloud Computing for Mobile World," 2010. (http://chetan.ueuo.com/projects/CCMW.pdf).
[7] Sun Microsystems, Inc., "Introduction to Cloud Computing Architecture", White Paper, 1st Edition, Jun. 2009.
[8] K. Kumar and Y.-H. Lu, "Cloud computing offloading for mobile users: Can offloading computation save energy?," IEEE Computer, no. 43-4, pp. 51-56, Apr. 2010.
[9] Koshy, K.I., "Sustainable Systems and Technology (ISSST)," 2012 IEEE International Symposium on, pp. 16-18 May 2012.
[10] Kamal Idrissi et al.,"A taxonomy and survey of Cloud computing," Security Days(JNS3), pp.

1-5, Apr. 2013.
〔11〕 E. Cuervo, A.Balasubramanian, D.ki Cho, A. Wolman, S.Saroiu, R.Chandra and P.Bahl. "Maui: making smartphones last longer with code offload," In Mobisys, pp. 49-62, Jun. 2010.
〔12〕 Ben Pfaff, Justin Pettit Teemu Koponen, Keith Amidon, Martin Casado, and Scott Shenker, "Extending Networking into th Virtualization Layer," HotNets-VII, pp. 22-23, Oct. 2009.
〔13〕 Xiaopeng Fan, Jiannong Cao, and Haixia Mao. "A Survey of Mobile Cloud Computing," ZTE Communications, no. 9-1, pp. 4-8, Mar. 2011.
〔14〕 Vikas Kottari et al, "A Survey on Mobile Cloud Computing: Concept,Applications and Challenges," Wireless Communication and Mobile Computing. no. 2, pp. 487-492, Mar. 2013.
〔15〕 H.S Abdelsalam, K.Marly, R.Mukkamala, M.Zubair and D.Kaminsky. "Analysis of energy efficiency in clouds," In COMPUTATIONWORLD'09:Proceedings of the 2009 computation world: future computing Service Computatio, Congnitive, Adaptive, Content, Patterns, pp. 416-421, Nov. 2009.
〔16〕 Andreas Berl, Erol Genlenbe, Marco DI Gelenbe, Giovanni Giuliani, Hermann DE Meer, Minh Quan Dang and Kostas Pentikousis, "Energy efficent cloud computing," The computer journal, Aug. 2009.
〔17〕 IEEE "Recommended Practice for Multi-Vendor -Access Point Interoperability via an Inter -Access Across Distribution Systems Supporting IEEE 802.1 Operation," IEEE Statndard 802.11, 2003.
〔18〕 Willem Vereecken, Lien Deboosere, Pieter Simoens, Brecht Vermeulen, Didier Colle, Chris Develder, Mario Pickavet, Bart Dhoedt, Piet Demeester. "Power efficiency of thin clients,", Eur.Trans.Telecomms. 2010.
〔19〕 Kiran Koshy, Andrew Juby, Vinod Namboodiri and Michael Overcash, "Can Cloud Computing Lead to Increased Sustainability of Mobile Devices?," Proceedings of 2012 IEEE International Symposium on Sustainable Systems and Technology (ISSST), pp. 16-18, May 2012.
〔20〕 Tae Hoon Keum et al., "A Performance Analysis Based on Hadoop Application's Characteristics in Cloud Computing,", Journal of The Korea Society of Computer and Information , no. 15-5, pp. May. 2010.

## 저 자 소 개

김 기 영
1996: 상지대학교
전자계산학과 공학사
1995: 삼보정보통신
기술연구소 연구원
1999: 숭실대학교
컴퓨터학과 공학석사
2003: 숭실대학교
컴퓨터학과 공학박사
현 재: 서일대학교
컴퓨터소프트웨어과 부교수
관심분야: 모바일컴퓨팅,센서네트워크, ITS, 네트워크보안
Email : ganet89@seoil.ac.kr

염 세 훈
1992: 국립 서울 산업대학교
전자계산학과 공학사
1994: 숭실대학교
컴퓨터학과 공학석사
2005: 숭실대학교
컴퓨터학과 공학박사
현 재: 동서울대학교
컴퓨터소프트웨어과 부교수
관심분야: 컴파일러, 프로그래밍언어, HCI, Voice Based Markup Language, XML
Email : shyeom@dsc.ac.kr